The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ



I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated 27 September 2001

Patent Form 1/77

Patents Act 1977
(Rule 16)

The Patent Office

12 DEC 2000

D Y & CO

THE PATENT OFFICE
12 DEC 2000
RECEIVED BY

# Request for a grant of a patent

*(See the notes on the back of this form you can also get an explanatory leaflet from the Patent Office to help you fill in this form)*

The Patent Office
Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference — P/10077GB

2. Patent application number *(The Patent Office will fill in this part)* — **0030260.4**

13DEC00 E590895-5 D02246
P01/7700 0.00-0030260.4

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

ARM Limited
110 Fulbourn Road
Cherry Hinton
Cambridge
CB1 9NJ
United Kingdom

Patents ADP number *(if you know it)* 74981240002.

If the applicant is a corporate body, give the country/state of its incorporation — United Kingdom

4. Title of the invention — Exclusive Access Control To A Processing Resource

5. Name of your agent *(if you have one)* — **D YOUNG & CO**

"Address for service" in the United Kingdom to which all correspondence should be sent (including the postcode) — **21 NEW FETTER LANE**
**LONDON**
**EC4A 1DA**

Patents ADP number *(if you know it)* — **59006**

6. If you are declaring priority from one or more earlier patent applications, give the country and date of filing of the or each of these earlier applications and *(if you know it)* the or each application number

| | Country | Priority application number *(if you know it)* | Date of filing *(day/month/year)* |
|---|---|---|---|
| 1st | | | |
| 2nd | | | |
| 3rd | | | |

7. If this application is divided or otherwise derived from an earlier UK application, give the number and filing date of the earlier application

| Number of earlier application | Date of filing (day/month/year) |
|---|---|
| | |

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
*a) any applicant named in part 3 is not an inventor, or*
*b) there is an inventor who is not named as an applicant, or*
*c) any named applicant is a corporate body.*
*See note (d))*

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

| | |
|---|---|
| Continuation sheets of this form | 0 |
| Description | 9 |
| Claim(s) | 4 |
| Abstract | 1 |
| Drawing(s) | 5 |

10. If you are also filing any of the following, state how many against each item

| | |
|---|---|
| Priority Documents | 0 |
| Translation of Priority Documents | 0 |
| Statement of inventorship and right to grant of a patent *(Patents Form 7/77)* | 2 |
| Request for preliminary examination and search *(Patents Form 9/77)* | 1 |
| Request for substantive examination *(Patents Form 10/77)* | 0 |
| Any other documents *(Please specify)* | 0 |

11. I/We request the grant of a Patent on the basis of this application.

Signature D Young & Co — Date 12 Dec 2000

**D YOUNG & CO**
Agents for the Applicants

12. Name and daytime telephone number of person to contact in the United Kingdom — Nigel Robinson — 023 80719500

**Warning**

*After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.*

**Notes**

*a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 01645 500505.*

*b) Write your answers in capital letters using black ink or you may type them.*

*c) if there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s) Any continuation sheets should be attached to this form.*

*d) If you answered 'Yes' Patents Form 7/77 will need to be filed.*

*e) Once you have filled in the form you must remember to sign and date it.*

*f) For details of the fee and ways to pay please contact the Patent Office.*

Pat n[illegible] rm 7/77

Patents Act 1977
(Rule 15)

12 DEC 2000

The Patent Office

THE PATENT OFFICE
D
12 DEC 2000
LONDON

D Y & CO

# Statement of inventorship and of right to grant of a patent

Th Pat nt Office
Cardiff Road
Newport
Gwent NP9 1RH

| | | |
|---|---|---|
| 1. | Your reference | P10077GB |
| 2. | Patent application number *(if you know it)* | **0030260.4** |
| 3. | Full name of the or of each applicant | ARM Limited |
| 4. | Title of the invention | Exclusive Access Control To A Processing Resource |
| 5. | State how the applicant*(s)* derived the right from the inventor*(s)* to be granted a patent | By Virtue of Employment |
| 6. | How many, if any, additional Patents Forms 7/77 are attached to this form? *(see note (c))* | 0 |

7. **I/We believe that the person*(s)* named over the page *(and on any extra copies of this forms)* is/are the inventor*(s)* of the invention which the above patent relates to.**

Signature Date

**D YOUNG & CO** 12 Dec 2000
Agents for the Applicants

| | | | |
|---|---|---|---|
| 8. | Name and daytime telephone number of person to contact in the United Kingdom | 023 80634816 | Nigel Robinson |

**Notes**

*a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

*b) Write answers in capital letters using black ink or you may type them.*

*c) If there are more than three inventor, please write the names and addresses of the other inventors on the back of another Patents Form 7/77 and attach it to this form.*

*d) When an application does not declare any priority, or declares priority from an earlier UK application, you must provide enough copies of this form so that the Patent Office can send one to each inventor who is not an applicant.*

*e) Once you have filled in the form you must remember to sign and date it.*

D Young & Co ref: P10077GB

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

| | |
|---|---|
| Surname | GRISENTHWAITE |
| First Names | Richard Roy |
| Address | 2 Connor's Close<br>Guilden Morden<br>Nr Royston<br>Cambridgeshire<br>SG8 0PT |
| Patents ADP number (if you know it): | 8041790001 |

| | |
|---|---|
| Surname | |
| First Names | |
| Address | |
| Patents ADP number (if you know it): | |

| | |
|---|---|
| Surname | |
| First Names | |
| Address | |
| Patents ADP number (if you know it): | |

**Reminder:**
**Have you signed the form?**

# EXCLUSIVE ACCESS CONTROL TO A PROCESSING RESOURCE

This invention relates to data processing systems. More particularly, this invention relates to data processing systems having control mechanisms for regulating exclusive access to processing resources.

It is known within data processing systems to provide processing resources, such as data elements within a database, that may be shared between different processors or processes. As an example, an airline reservation database system may have a central database storing information regarding the current booking status of an aircraft. This central database may be accessed from many different computers independently and may also be accessed by different processes running on a single computer. In order to ensure the integrity of the data concerned, exclusive access control mechanisms are used whereby exclusive access to a certain portion of the database is given to a particular processor or process at any one time to ensure that different copies of the same data do not come into existence causing data integrity problems.

It is known to use semaphore values associated with a processing resource to indicate whether exclusive access to that processing resource is available to an access requester. More particularly, a read instruction may read a semaphore value for the purpose of determining whether or not exclusive access may be granted. If the semaphore value indicates that exclusive access would be available, then another instruction is executed to modify the semaphore value to indicate that exclusive access has been given to the access requester.

Whilst such an arrangement can regulate the access to processing resources, real life systems also need to be able to accommodate such a mechanism within a system that may be subject to the occurrence of interrupts, exceptions or context switches, for example, that may intervene between a semaphore value being read to determine if exclusive access is available and the semaphore value being written to indicate that exclusive access has been allowed. The need to allow for such occurrences whilst not unduly impacting the latency of the system presents a technical problem.

Viewed from one aspect the present invention provides a method of processing data, said method comprising the steps of:

retrieving a semaphore value corresponding to a processing resource from a semaphore value store;

storing semaphore identifying data indicative of which semaphore value has been retrieved;

determining from said semaphore value whether or not said processing resource is available for exclusive access by a requesting exclusive access requestor; and

writing a new semaphore value to said semaphore value store, said new semaphore value being indicative of exclusive access being granted to said exclusive access requestor; wherein

in response to execution of an exclusive access clear instruction by an exclusive access requestor clearing stored semaphore identifying data for said exclusive access requestor.

The invention provides an exclusive access clear instruction that serves to clear any semaphore identifying data that has been stored between retrieving a semaphore value and writing a new semaphore value such that a fresh start to establishing an exclusive access permission to a processing resource may later be forced to occur thereby avoiding problems due to intervening processing. As a particular example, should an interrupt, exception, or context switch occur, then an exclusive access clear instruction may be executed by the operating system so as to flush out any pending requests for exclusive access that have not yet been granted and properly locked in place thereby avoiding improper operation.

In order to accommodate multiple exclusive access requests from different sources, the step of writing a new semaphore value returns a result value indicative of whether or not that new semaphore value was written. The result value allows a determination to be made as to whether or not some other processor or process has been granted exclusive access to the processing resource before the step of writing for the current access requester was able to establish the exclusive access for that access requester.

The semaphore identifying data could be used in a variety of ways to control the establishing of exclusive access permissions, but a particularly efficient way of providing safe control is to arrange for the step of writing the new semaphore value to check the semaphore identifying data to determine whether or not it has been cleared between the step of storing and the step of writing.

In the case that an interrupt, exception, context switch or some other event had occurred between the storing and the subsequent write attempt, then an exclusive access clear instruction will have been executed to clear the semaphore identifying data and accordingly the write attempt will not succeed and an inappropriate exclusive access permission not granted. Avoiding an inappropriate write may also save bus resources within a system having a shared bus via which the write action must be performed, e.g. if the write action will use the main memory, which is typically on a shared bus, the stopping this write saves bus resources. Freeing the shared bus for use by other parts of the system increases the overall efficiency of the system.

It will be appreciated that the exclusive access requestors could be different processors within a multiprocessor system, or different tasks/processes within a multitasking/multiprocessing system.

The semaphore identifying data may be stored locally to the exclusive access requester, local to the shared processing resource, or in both places. Storing the semaphore identifying data locally increases the speed with which this may be accessed and reduces the amount of resources that need to be provided to give non-local access to an exclusive access requestor.

It will be appreciated that the processing resource could take a variety of different forms. For example, the processing resource could be an input/output port. A high level memory device or the like. However, the invention is particularly useful in the control of exclusive access to data elements within a data memory.

Viewed from another aspect the invention provides an apparatus for processing data, said apparatus comprising:

retrieving logic operable to retrieve a semaphore value corresponding to a processing resource from a semaphore value store;

storing logic operable to store semaphore identifying data indicative of which semaphore value has been retrieved;

determining logic operable to determine from said semaphore value whether or not said processing resource is available for exclusive access by a requesting exclusive access requestor; and

writing logic operable to write a new semaphore value to said semaphore value store, said new semaphore value being indicative of exclusive access being granted to said exclusive access requestor; wherein

in response to execution of an exclusive access clear instruction by an exclusive access requestor, clearing logic is operable to clear stored semaphore identifying data for said exclusive access requestor.

The invention may also be embodied as a computer program product bearing a computer program for controlling a data processing apparatus in accordance with the above-described techniques.

An embodiment of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 schematically illustrates a multiprocessor system having a shared main memory to which exclusive memory access requests may be made;

Figure 2 is a flow diagram illustrating the normal operation of establishing an exclusive access permission within the system of Figure 1;

Figure 3 is an example of some ARM processor object code that may be used to establish an exclusive access permission;

Figure 4 illustrates a possible interaction between the two processors of Figure 1 competing to obtain exclusive access to the same data element within the memory; and

Figure 5 is a flow diagram illustrating the action of an exclusive access clear instruction within the system of Figure 1.

Figure 1 schematically illustrates a data processing system 2 including a first processor 4 and a second processor 6. A first cache memory 8 is associated with the first processor 4 and a second cache memory 10 is associated with the second processor 6. A shared address and data bus 12 connects the first processor 4 and the second processor 6 with other system elements including a UART circuit 14, a DMA circuit 16 and a shared main memory 18. A bus arbiter 20 serves to ensure that only one bus master may have control of the shared bus 12 at any given time.

The main memory 18 stores data elements, such as for example database records, to which it is required to allow exclusive access by one of the first processor 4 and the second processor 6. Semaphore values are stored within the main memory 18 for controlling the access to data elements, such as individual data records or areas of memory, for which exclusive access requests may be made. Monitoring circuitry 22 is provided local to the main memory 18 for storing data identifying which semaphore values have been retrieved by which processor 4 or 6. The semaphore identifying data takes the form of the physical memory address of the semaphore within the main memory together with processor identifying data, such as a processor number allocated by the bus arbiter 20. A semaphore identifying data store 24 is associated with the cache memory 8 and a further semaphore identifying data store 26 is associated with the cache memory 10. These semaphore data identifying stores 24, 26 are provided local to their respective processors 4, 6. When a retrieve operation is performed for a semaphore value within the main memory 18, the physical address of that semaphore value is stored within the respective local semaphore identifying data store 24, 26. Only a single pending exclusive access establishing operation is allowed to exist for each processor 4, 6 at any given time. The two processors 4, 6 may each have their own pending exclusive access establishing request, but the software programmers of the data processing system 2 are constrained not to rely upon more than one pending exclusive access requesting operation within a given processor 4, 6.

The processors 4, 6 may operate using virtual addresses, but a translation to physical addresses using a translation lookaside buffer (TLB) occurs within each

processor 4, 6 before the memory addresses are output to the respective cache memories 8, 10 and the main memory 18. Thus, the semaphore identifying data relates to physical addresses rather than virtual addresses in this embodiment.

Figure 2 is a flow diagram schematically illustrating the operation of the system of Figure 1. At step 26, one of the processors 4, 6 executes an LDREX instruction that loads a semaphore value from a memory location (specified within a register $R_{m)}$ within the main memory 18. The LDREX instruction is different from a standard ARM LDR instruction that is used to load a memory value. The LDREX instruction utilises the mechanisms for establishing exclusive access relationships whereas the standard LDR instruction does not.

At step 28, a check is made as to whether or not the semaphore value being sought is cacheable. If the semaphore value is not cacheable, as indicated within an associated MMU (not illustrated), then processing proceeds to step 30. If the semaphore value is cacheable, then processing proceeds to step 32.

At step 30, the semaphore value corresponding to the address indicated by the $R_m$ register value is stored into a register $R_d$ specified within the LDREX instruction. Furthermore, the physical address of the semaphore value together with a processor identifying number for the processor executing the LDREX instruction is stored within the main memory monitor circuit 22. The bus arbiter 20 is triggered to provide this processor identifying number to the main memory monitor circuit 22 by the decoding of an LDREX instruction by one of the processors 4, 6. At step 30, the physical address of the semaphore data is also stored within the local semaphore identifying data store 24, 26 of the processor executing the LDREX instruction.

If processing from step 28 proceeds to step 32, instead of step 30, then no data is written into the main memory monitor circuit 22, but the physical address of the semaphore value being returned from address $R_m$ to register $R_d$ is written into the appropriate local semaphore identifying data store 24, 26.

At step 34, the semaphore value returned from the main memory 18 to one of the processors 4, 6 is examined to determine whether the retrieved semaphore value

indicates that exclusive access to the associated data element is permitted. If exclusive access is not permitted, then processing terminates. Effectively, the exclusive access request attempt will be retried at some later time, possibly immediately using a tightly looped portion of code.

If the returned semaphore value indicates at step 36 that exclusive access is possible, then processing proceeds to step 38 at which an STREX instruction is executed. The STREX instruction seeks to store a new semaphore value into the main memory 18 indicating the exclusive access permission that has been granted. The STREX differs from a standard ARM STR instruction in that an additional check on the pending semaphore identifying data is made and a result value is returned indicating whether or not the write was completed.

If the write was not completed, then this is indicative of another processor or process having established an exclusive access permission to the same data element in the intervening time between reading the semaphore value at step 26 and attempting to execute a corresponding STREX instruction at step 38.

If the physical memory address of the semaphore value indicates that the semaphore value is cacheable then processing proceeds to step 40. If the semaphore value is non-cacheable, then processing proceeds to step 42.

Step 42 checks within the local semaphore identifying data store 24, 26 and the main memory monitor circuit 22 that the physical address of the semaphore value that is being written to is still stored (i.e. has not been cleared or overwritten), and in the case of the main memory monitor circuit 22 that the matching processor number is associated with the physical memory address. The check is first made with the local store 24, 26. If this has been cleared, then a fail result is returned, otherwise the main memory monitor circuit 22 is examined and the result returned from there. If both these conditions are not met, then processing proceeds to step 44 at which a fail result value is returned from the STREX instruction, the write not having taken place and processing terminates. The exclusive access request attempt may in practice be retried.

If the tests of step 42 indicated that the semaphore identifying data stored both locally to the processor 4, 6 and within the main memory monitor circuit 22 still match (i.e. there has been no intervening exclusive access permission granted to another process, an intervening clear instruction or some other interference with the normal exclusive access permission granting process), then processing proceeds to step 46 at which the semaphore value within the main memory is updated to indicate the granting of the exclusive access permission to the originator of the STREX instruction and a pass result value returned.

If the test as to whether or not the new semaphore value of the STREX instruction was cacheable indicated that it was cacheable, then step 40 checks within the local semaphore identifying data store 24, 26 that the matching physical address is still present. If the matching physical address is not still present, then processing proceeds to step 48 at which a fail result value is returned. If the matching result is still present, then processing proceeds to step 50 at which the semaphore value within the main memory is updated and a pass result value returned.

Figure 3 schematically illustrates an ARM object code routine for establishing an exclusive access permission. The LDREX instruction returns a semaphore value from the main memory 18 and sets up the appropriate semaphore identifying data. The CMP instruction determines whether or not the semaphore value indicates that exclusive access is possible. If exclusive access is not possible, then the BNE instruction returns processing to retry the request. If exclusive access is possible, then processing proceeds to further instructions, not illustrated, that set up the desired new semaphore value in register $R_d$. The instruction STREX serves to attempt to write the new semaphore value to the main memory 18. In accordance with the operation described in relation to Figure 2, the STREX instruction will only properly complete if the semaphore identifying data in the appropriate stores matches indicating that no inappropriate intervening action has occurred that would interfere with a proper exclusive access relationship being established. The CMP instruction following the STREX instruction examines the result value returned from the STREX instruction to determine whether or not the new semaphore value was properly stored and accordingly that the exclusive access relationship was established. The BNE instruction again attempts to retry the process if the result value indicated a fail.

Figure 4 schematically illustrates two processors that are competing to establish an exclusive access relationship to the same data elements. Processor 1 issues its LDREX instruction to read the semaphore first. This indicates that an exclusive access permission is possible. However, before Processor 1 can issue its STREX instruction to confirm that exclusive access relationship, Processor 2 both reads and writes the same semaphore value to establish an exclusive access relationship for Processor 2. Accordingly, when the STREX instruction for Processor 1 is attempted, this returns a fail value indicating that the exclusive access permission is not possible even though the semaphore value read by Processor 1 had indicated that this was a possibility.

Figure 5 schematically illustrates the operation of a CLREX instruction. This CLREX instruction is executed as an early step within any interrupt code or exception code as well as by context switching control software within a multitasking environment. The CLREX instruction serves to clear out any pending exclusive access permission requests that may be present for the processor executing the CLREX instruction. In practice, since the semaphore identifying data is stored within the local semaphore identifying data store 24, 26 of the processor 4, 6 irrespective of whether or not the semaphore value is cacheable, then clearing this local semaphore identifying data for the processor has the effect of stopping subsequent STREX instructions executing inappropriately for that semaphore value. In addition, providing this check, locally to the processor 4, 6 avoids the use of the shared system bus 12 thereby releasing this resource for use by other elements within the data processing system 2.

## CLAIMS

1. A method of processing data, said method comprising the steps of:

retrieving a semaphore value corresponding to a processing resource from a semaphore value store;

storing semaphore identifying data indicative of which semaphore value has been retrieved;

determining from said semaphore value whether or not said processing resource is available for exclusive access by a requesting exclusive access requestor; and

writing a new semaphore value to said semaphore value store, said new semaphore value being indicative of exclusive access being granted to said exclusive access requestor; wherein

in response to execution of an exclusive access clear instruction by an exclusive access requestor, clearing stored semaphore identifying data for said exclusive access requestor.

2. A method as claimed in claim 1, wherein said step of writing a new semaphore value returns a result value indicative of whether or not said new semaphore value was written in said semaphore value store.

3. A method as claimed in claim 2, wherein if a different exclusive access requestor has written a new semaphore value to said semaphore value store between said step of retrieving and said step of writing, then said result value indicates that said write of a new semaphore value by said exclusive access requestor has failed.

4. A method as claimed in any one of claims 1, 2 and 3, wherein said step of writing also checks said semaphore identifying data to determine whether or not said semaphore identifying data has been cleared between said step of retrieving and said step of writing.

5. A method as claimed in claim 4, wherein, if said semaphore identifying data has been cleared, then writing of said new semaphore value is not attempted.

6. A method as claimed in any one of the preceding claims, wherein a plurality of data processors share said processing resource.

7. A method as claimed claim 6, wherein said plurality of data processors share at least a common access point via which accesses to said processing resource are made.

8. A method as claimed in any one of the preceding claims, wherein a local semaphore identifying data store is provided local to said exclusive access requestor.

9. A method as claimed in claims 7 and 8, wherein a write attempt does not reach said common access point if said semaphore identifying value stored in said local semaphore identifying data store has been cleared.

10. A method as claimed in any one of the preceding claims, wherein a shared semaphore identifying data store is provided local to said processing resource.

11. A method as claimed in any one of the preceding claims, wherein multitasking processing is performed such that different processing tasks may act as different exclusive access requestors.

12. A method as claimed in claims 5, 8 and 11, wherein said exclusive access clear instruction clears said local semaphore identifying data store, but not said shared semaphore identifying data store, and said semaphore identifying data within said local semaphore identifying data store is checked to determine whether or not said semaphore identifying data has been cleared between said step of retrieving and said step of writing.

13. A method as claimed in any one of the preceding claims, wherein said processing resource is a data element stored within a data memory.

14. A method as claimed in any one of the preceding claims, wherein an exclusive access clear instruction is executed upon occurrence of one or more of:

an exception triggering exception handling; and

a context switch between different tasks within multitasking operation.

15. A method as claimed in any one of the preceding claims, wherein said semaphore identifying data is data indicative of a memory address associated with said processing resource.

16. A method as claimed in claims 6 and 10, wherein said shared semaphore identifying data store stores data indicative of which processor is requesting exclusive access to said processing resource.

17. A computer program product carrying a computer program for controlling a data processing apparatus in accordance with the method of any one of claims 1 to 16.

18. Apparatus for processing data, said apparatus comprising:

retrieving logic operable to retrieve a semaphore value corresponding to a processing resource from a semaphore value store;

storing logic operable to store semaphore identifying data indicative of which semaphore value has been retrieved;

determining logic operable to determine from said semaphore value whether or not said processing resource is available for exclusive access by a requesting exclusive access requestor; and

writing logic operable to write a new semaphore value to said semaphore value store, said new semaphore value being indicative of exclusive access being granted to said exclusive access requestor; wherein

in response to execution of an exclusive access clear instruction by a exclusive access requestor, clearing logic is operable to clear stored semaphore identifying data for said exclusive access requestor.

19. A method of processing data substantially as hereinbefore described with reference to the accompanying drawings.

20. A computer program product carrying a computer program for controlling a data processing apparatus substantially as hereinbefore described with reference to the accompanying drawings.

21. Apparatus for data processing substantially as hereinbefore described with reference to the accompanying drawings.

## ABSTRACT

## EXCLUSIVE ACCESS CONTROL TO A PROCESSING RESOURCE

A data processing system (2) is provided with multiple processors (4, 6) that share a main memory (18). Semaphore values associated with data elements within the memory system, including the main memory (18), are used to establish exclusive access permissions to those data elements. An exclusive access clear instruction (CLREX) is provided that serves to clear any partially completed exclusive access requests for a processor between the reading of a semaphore value and the writing of a semaphore value to establish the exclusive access permission.

[Figure 2]

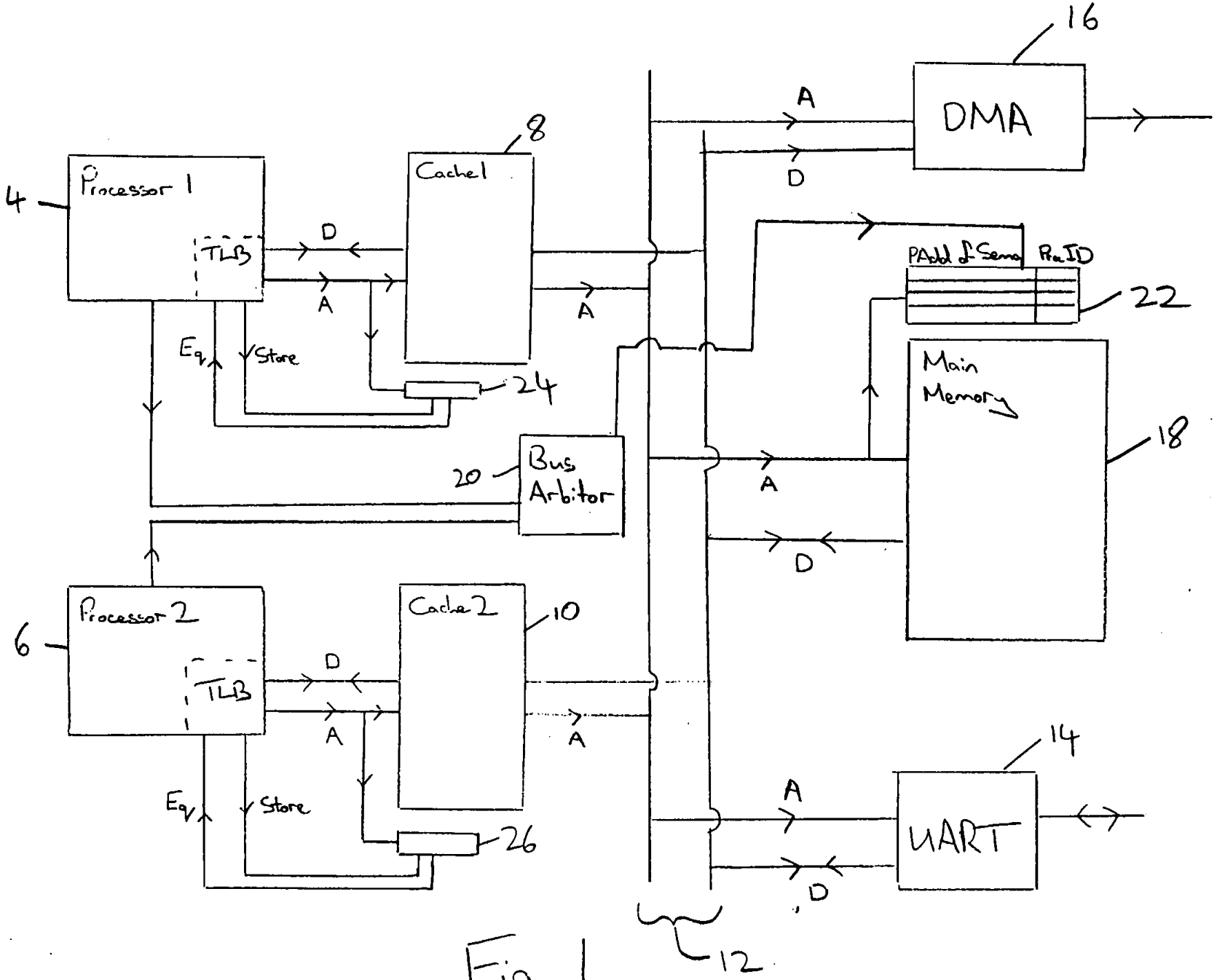
16
2
DMA
A
D
8
Cache1
4
Processor 1
TLB
D
A
A
Eq
Store
24
PAdd
Seno
ReID
22
Main
Memory
18
20
Bus
Arbitor
A
D
Processor 2
6
Cache2
10
TLB
D
A
A
Eq
Store
26
14
UART
A
D
12

Fig. 1

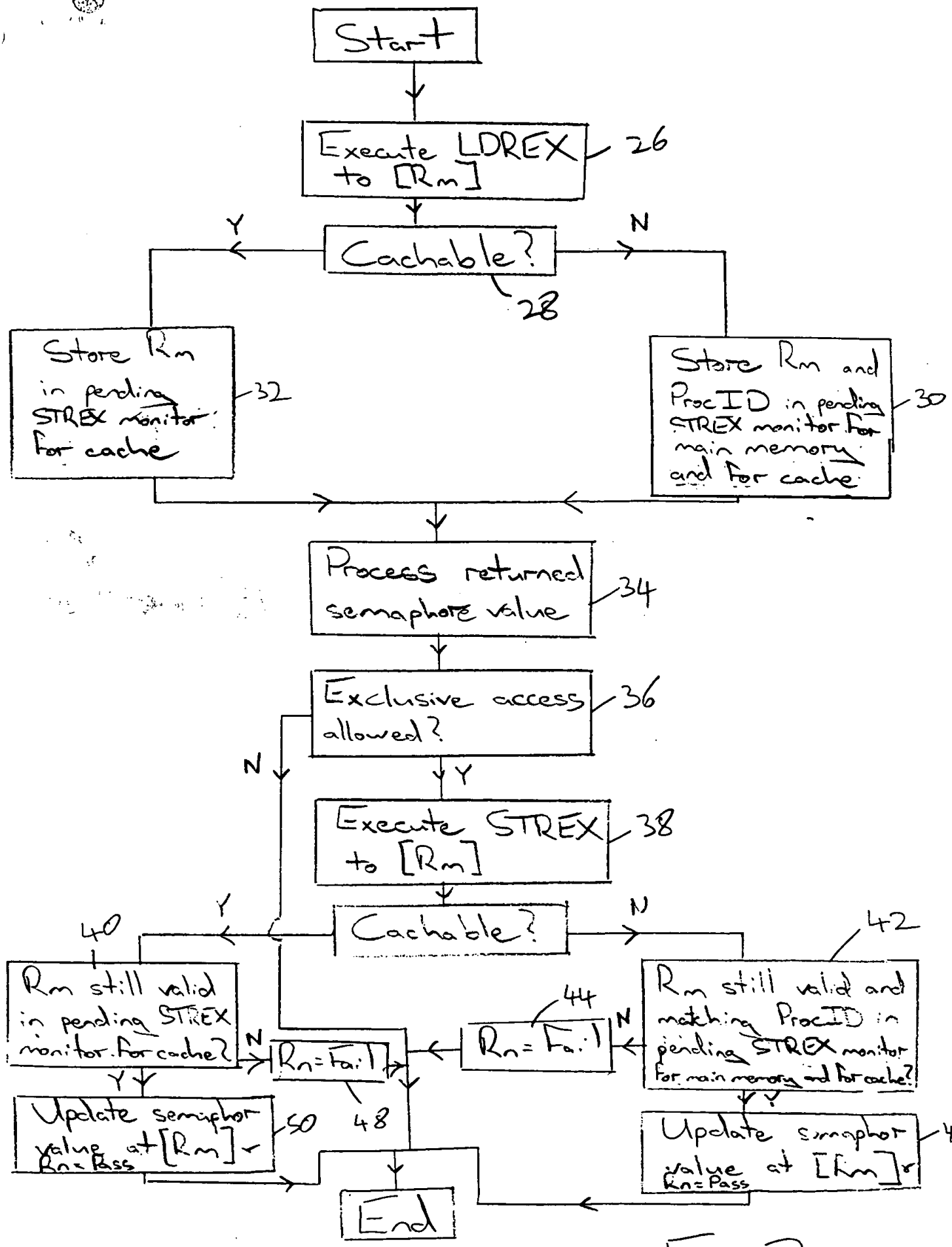
Start
Execute LDREX to [Rm]
26
Cachable?
28
Y
N
Store Rm in pending STREX monitor for cache
32
Store Rm and ProcID in pending STREX monitor for main memory and for cache
30
Process returned semaphore value
34
Exclusive access allowed?
36
N
Y
Execute STREX to [Rm]
38
Cachable?
Y
N
40
42
Rm still valid in pending STREX monitor for cache?
Rm still valid and matching ProcID in pending STREX monitor for main memory and for cache?
44
Rm = Fail
Rm = Fail
N
N
Y
Y
Update semaphor value at [Rm] Rm = Pass
50
48
Update semaphor value at [Rm] Rm = Pass
End

Fig. 2

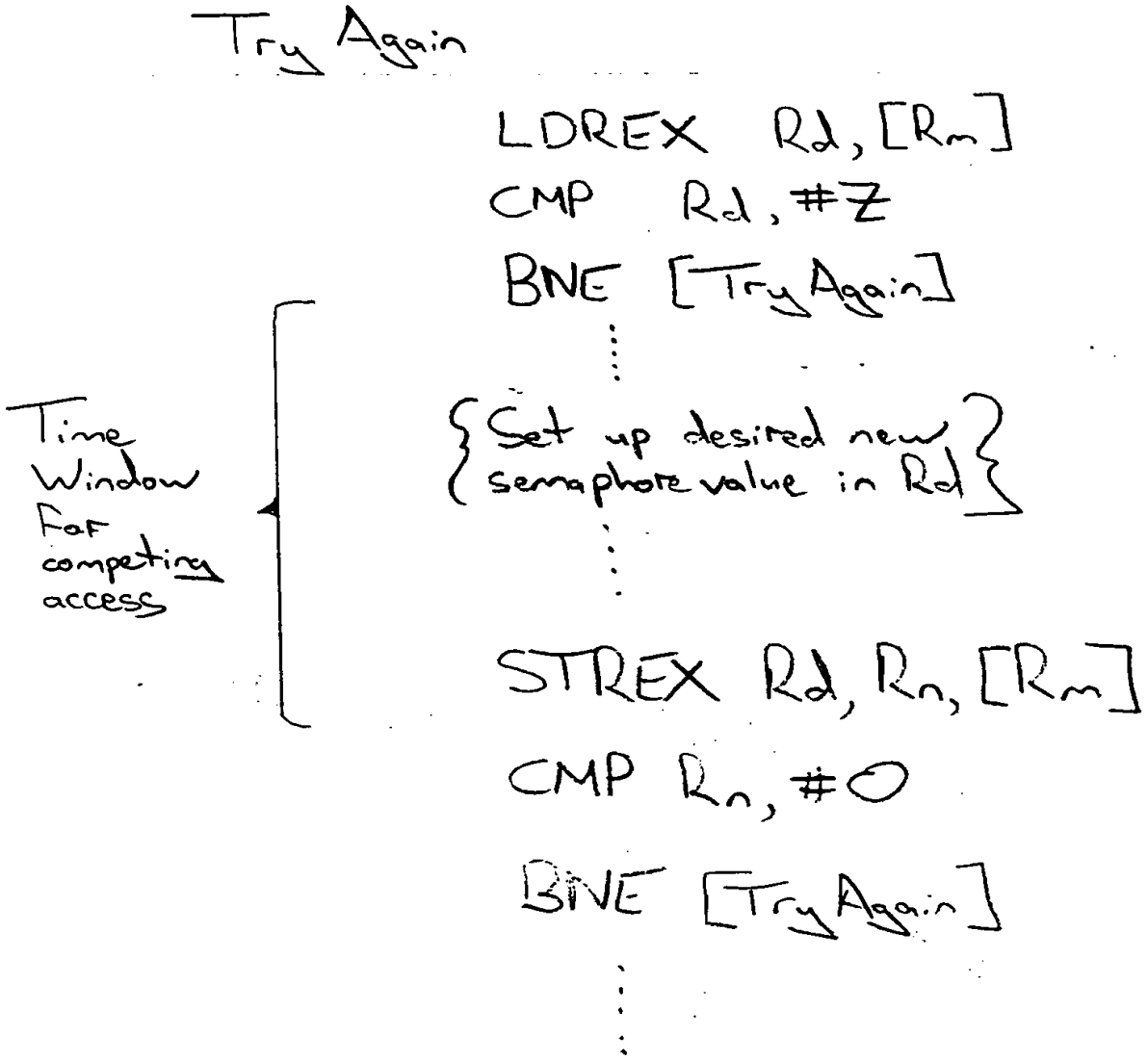
Try Again
LDREX Rd, [Rm]
CMP Rd, #Z
BNE [Try Again]
{Set up desired new semaphore value in Rd}
Time Window For competing access
STREX Rd, Rn, [Rm]
CMP Rn, #0
BNE [Try Again]

Fig. 3

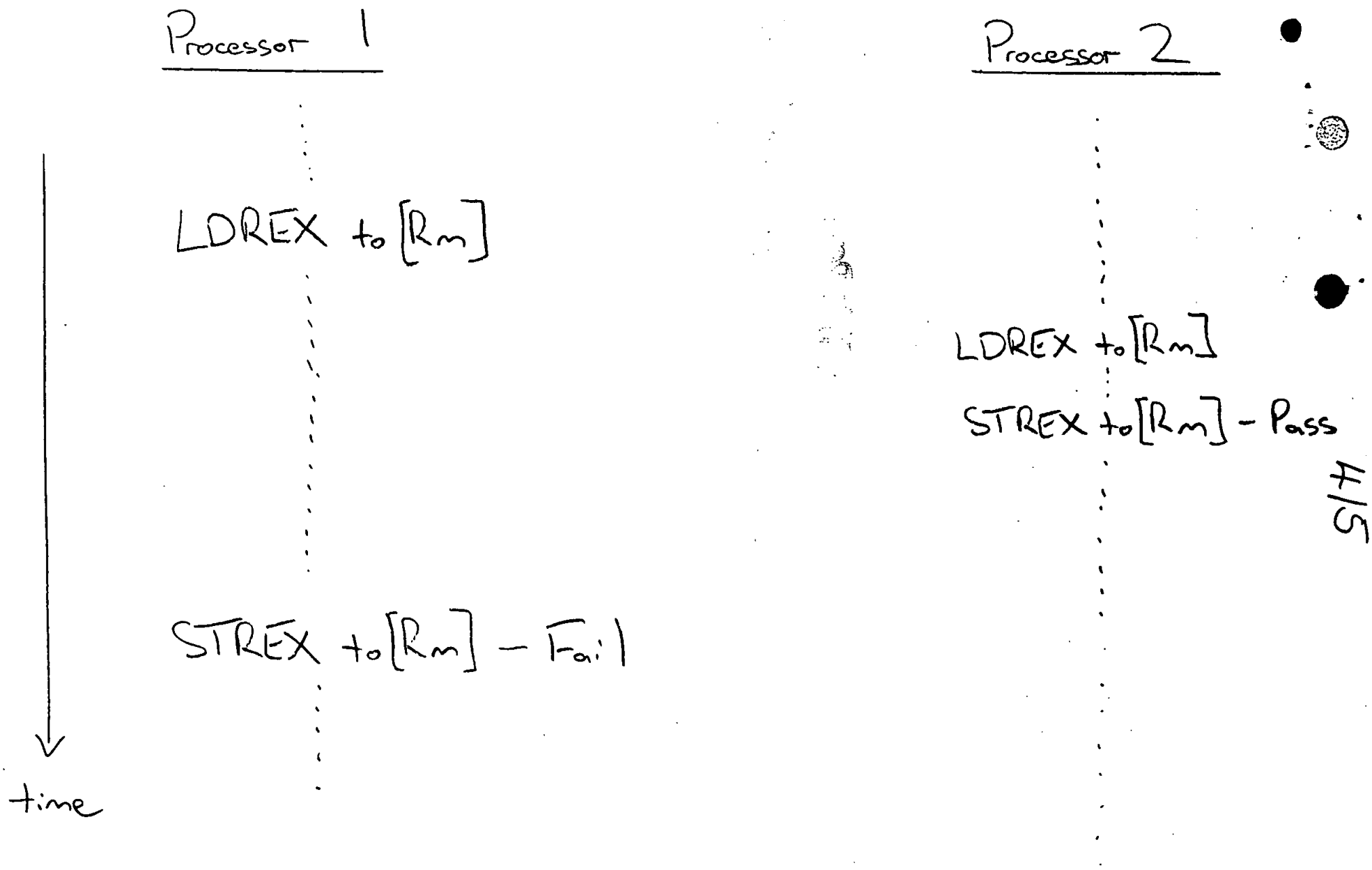
Processor 1
Processor 2
LDREX to [Rm]
LDREX to [Rm]
STREX to [Rm] - Pass
STREX to [Rm] - Fail
time

Fig. 4

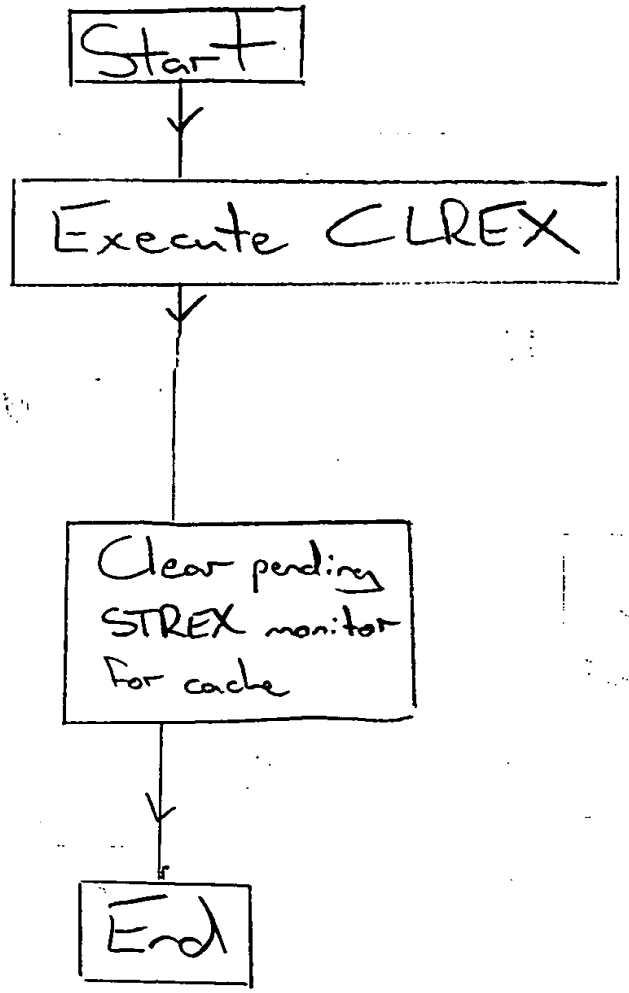
Start
Execute CLREX
Clear pending STREX monitor for cache
End

Fig. 5